Bachelor's thesis

Business Information Technology

Business Data Communications and Information Security

2015

Laura Kankaala

IDENTITY FEDERATION USING SHIBBOLETH IDENTITY PROVIDER

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Business Data Communications & Information Security

2015 | 42 pages

Jarkko Paavola (Turun Ammattikorkeakoulu) and Ilkka Heimo (KPMG)

Laura Kankaala

IDENTITY FEDERATION USING SHIBBOLETH IDENTITY PROVIDER

The employer of this thesis is a global company called KPMG. In addition to KPMG's primary accounting services, the company has also expanded its expertise to the field of information security. The purpose of this thesis was to construct an environment for KPMG to showcase identity federation solutions, using Shibboleth Identity Provider.

The research method used in this thesis is constructive and it is based on a real life use case. Some of the references used in this thesis are from the environment that was built, while most of the facts are based on scientific articles and studies

Objective was to use Shibboleth Identity Provider software, because it is used by many major organisations as a foundation of their identity federation services. This thesis covers the basic steps of establishing an identity federation environment, as well as discusses the different options and viewpoints for choosing a specific platforms or software.

Identity federation stands for the means and standards of transferring user-related information from one security domain to another through an interface such as a web browser, allowing companies or organisations to separate user authentication to a separate service. The purpose of adapting an identity federation system is to achieve economic benefits, better user experience and enhance information security.

In conclusion, the identity federation environment that was built is a very generic and has room for future improvements. However, it demonstrates and describes its primal purpose of transferring user identifying information across different systems.

KEYWORDS:

Identity Federation, Access Management, Shibboleth, Apache

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Yrityksen tietoliikenne ja tietoturva

2015 | 42 sivua

Jarkko Paavola (Turun Ammattikorkeakoulu) ja Ilkka Heimo (KPMG)

Laura Kankaala

IDENTITEETIN FEDEROINTI KÄYTTÄEN SHIBBOLETH IDENTITY PROVIDER -OHJELMISTOA

Opinnäytetyön toimeksiantajana on globaali yritys nimeltään KPMG. Alunperin ainoastaan tilintarkastuspalveluita tarjoava KPMG on sittemmin laajentanut repertuaariaan kattamaan myös tietoturvapalveluita. Opinnäytetyön tarkoituksena oli rakentaa KPMG:lle ympäristö, jolla voidaan esitellä identiteetin federoinnin ominaisuuksia. Identiteetin federointi voidaan ilmaista myös termillä luottamusverkko.

Opinnäytetyön tutkimusmenetelmä on konstruktiivinen ja se perustuu tosielämän käyttötapaukseen. Osa lähteistä on suoraan peräisin rakennetusta ympäristöstä, mutta suurin osa tiedosta perustuu moniin eri tieteellisiin artikkeleihin ja tutkielmiin.

Tarkoituksena on käyttää Shibboleth Identity Provider ohjelmistoa, koska monet suuret yritykset käyttävät sitä luottamusverkkojen pohjana. Opinnäytetyö kattaa luottamusverkon luomisen periaatteet, sekä esittelee erilaisia vaihtoehtoja ja näkökulmia tiettyjen ohjelmistojen tai alustojen valitsemisien tueksi.

Luottamusverkolla tarkoitetaan tapoja ja standardeja, joilla siirretään käyttäjään liittyvää tietoa eri tietoturvaympäristöstä toiseen, mahdollistaen sen, että organisaatiot tai yritykset voivat erottaa käyttäjän tunnistamisen erilliseksi palveluksi. Luottamusverkoilla tavoitellaan taloudellisia hyötyjä, parempaa käyttäjäkokemusta, sekä parannetaan tietoturvaa.

Lopputulemana on se, että opinnäytetyönä rakennettu ympäristö on hyvin geneerinen ja siihen jää kehittämisen varaa. Joka tapauksessa, se toteuttaa sen perustavanlaatuisen tarkoituksensa, eli onnistuu välittämään käyttäjän tunnistamiseen liityvää tietoa paikasta toiseen.

ASIASANAT:

Identiteetin federointi, luottamusverkko, pääsyhallinta, Shibboleth, Apache

TABLE OF CONTENTS

APPENDICES

PICTURES

LIST OF ABBREVIATIONS

ADFS	Active Directory Federation Services (Microsoft 2015).
API	Application Programming Interfaces (Apache Portable Runtime Project 2015).
AJP	Apache Jserv Protocol (Apache Tomcat 2015).
APR	Apache Portable Runtime (Apache Software Foundation 2015).
CA	Certificate Authority (Kyrnin 2007).
HTTP	Hypertext Transfer Protocol (Beal, 2015).
IAM	Identity and Access Management (Gaedke 2015).
IdP	Shibboleth Identity Provider software (Shibboleth 2015).
PCRE	Perl-Compatible Regular Expressions Library (Apache Software Foundation 2015).
SAML	Security Assertion Markup (OASIS 2008).
SP	Shibboleth Service Provider (Shibboleth 2015).
SSL	Secure Sockets Layer (OpenSSL Project 2015).
TLS	Transport Security Layer (OpenSSL Project 2015).
XML	Extensible Markup Language (W3 2015).

1 INTRODUCTION

The scope of this thesis is to provide an example of how identity federation can be achieved using Shibboleth Identity Provider (IdP) software, and discuss the requirements and prerequisites to complete this real-life use case. However, to understand the structure and the methods used, it is vital to provide a short theoretical section as a foundation for this thesis. More thorough explanation of the identity federation will be disclosed briefly below in the next chapter.

Prior to writing this thesis, I constructed an identity federation system using the Shibboleth IdP software and Apache's servers. The environment is to be a part of a proof of concept environment for a company called KPMG, to demonstrate benefits of federated access management for potential customers. KMPG is an international company, whose primary services were accounting services, but since then the company has expanded its expertise to the field of information security, especially to offer Identity and Access Management (IAM) solutions.

The end result contributes as a part of larger proof of concept environment, including also other aspects of access management, which are not disclosed in this thesis. I was motivated to take on this assignment, as it provided a great way to learn more of the field of my employment. The work I did gave me a chance to see the fundamental level of the systems I work with daily.

In addition to Shibboleth IdP, the environment in this thesis is built completely upon open source software. Apache HTTPD is used as a frontend server and Apache Tomcat as a backend server, together with OpenSSL to provide Secure Sockets Layer (SSL) and Transport Layer Security (TLS). The purpose of the aforementioned will be discussed in the third chapter. The thesis will walk the reader through the prerequisite steps of building an identity federation system and also present some plausible options for the platforms.

Fourth chapter focuses in depth on the subject of Shibboleth IdP. The aforementioned software is one part of Shibboleth's Identity Federation solution fami-

ly. Shibboleth is an open-source project and each component is open source and can be used free of charge even commercially (Shibboleth 2015).

2 IDENTITY FEDERATION AND IDENTITY AND ACCESS MANAGEMENT

Identity federation should be seen as a part of IAM architecture. IAM solutions are attempting to provide a solution for the amassing security threats that are posed by distributed and mobile computing, as well as overall increased usage of IT systems (Hudson 2011, 2). Sharing resources among heterogeneous organisations or IT systems is gratuitously complicated, as most systems deploy built-in identity and access control mechanisms. Therefore IAM solutions seek to develop technologies to separate authentication and authorization processes from the applications with the help of web-based technologies and standards. (Gaedke 2015, 1)

In addition to fulfilling different business needs, there are also legislative reasons that demand the organisations to protect digital identities. Finnish Ministry of Finance for example states access management registries are affected by the same legislation as personal data protection and care liabilities legislation (Finnish Ministry of Finance 1999/523). As such, databases that hold user information need to be protected, and only required information on the user may be collected, which makes carefully planned IAM increasingly crucial.

As mentioned above, identity federation system is tightly dependent on IAM, which refers to a compilation of standards, policies and technologies used to securely provide a user with a digital identity. Identity must be distinguishable from other digital identities in the given network and it consists of characteristic elements. These elements are called identifiers when they are used in the identification process, and commonly the identity is represented as a unique identifier, a value such as an account name or number (Josang 2015, 1). In other words, identifiers are used to determine the identity of a user. Digital identity is sometimes referred to as federated identity (OASIS 2008, 8), depending on the context and terminology used. The employer of this thesis has chosen the term digital identity over federated identity, which is why this term is used in this thesis.

Two or more organisations, or within one organization's borders, identity federation systems can be established to share IAM system. The digital identity attributes are distributed in agreed upon standards, rules and policies, to enable users to access protected resources in a secure and easy-to-use manner (EDUCAUSE 2015, 1). Identity federation systems must also conclude how the user is authenticated, for example how the validity and correctness of the user's unique identifier is verified (Josang 2015, 1-2).

Different standards are used to govern the exchange of the digital identity identifiers. The standard used by Shibboleth IdP is Security Assertion Markup Language (SAML) 2.0, which will be described in detail later in the thesis. In addition to SAML 2.0, OAuth, OpenID and Facebook Connect are widely used, albeit to satisfy different business cases and to function with different software or applications (Mehta 2014).

A concrete example of an identity federation system is Haka trust network, an identity federation system used by Universities and Universities of Applied Sciences in Nordic Countries, and University hospitals, research centres and the organisations that provide services for the aforementioned parties (Tuomi 2015). This enables students and staff to use only one set of username and password combination to access resources from other domains more efficiently. This kind of feature makes it also easier for administrators to add access rights for users, as they do not need to be recreated or replicated to another user repository.

3 PREREQUISITES AND PLATFORMS

Shibboleth IdP software requires certain platforms and components in order to function properly. The next sub-sections describe in detail Apache HTTPD, Apache Tomcat and OpenSSL that are the foundation upon which Shibboleth IdP is compiled. The whole environment is built on CentOS operating system, using one frontend and one backend. The same system could be produced with any number of previously mentioned elements. Similar environment could be built using package-managing utilities, such as Yum, which would streamline the process by automatically compiling the servers, but to comprehend the system as a whole, the building blocks are installed separately.

For security reasons, the following elements should not be installed or run as a root user. If any of the servers are compromised and they are run as a root, it means the attacker can directly exploit the whole system, such as its connections to different applications or databases (Wallen 2009). It is advisable to create new user group and user and give them recursive rights to write, read and execute only in required directories.

3.1 Apache HTTPD

The most logical way to go through the prerequisite platforms for the thesis is to start with the frontend Apache HTTPD server. This thesis uses release 2.4.12 of Apache HTTPD. This server was picked as it is open source and can be downloaded freely from Apache's website. Alternative option for a frontend server could be IBM's HTTP Server, which is also free of charge and is based on Apache's HTTPD server (IBM 2015). However, to avoid compatibility issues, IBM's HTTP Server should run with a chargeable IBM WebSphere Application server as a backend, which is why Apache HTTPD was chosen. As default, compatibility between products from different vendors cannot be guaranteed.

Frontend and backend servers are program interfaces, or layers, which provide content for the user. User can interact directly with frontend servers by web

browser or other interface (WhatIs 2006). This thesis uses the frontend server to collect user information, in other words, user provided username and password combination and relay them securely to Shibboleth IdP located at the backend server. The credentials can be requested an outside source, for example an application outside of this specific system that can validate the user through the IdP.

The HTTPD server is first uploaded to the server using the *wget* command, which is used to download content from websites via the command line interface. After the installation, certain modifications need to be done in order to achieve the basic login form that collects credentials from the user and relays them to the IdP.

3.1.1 Compiling Apache HTTPD

Prior to compiling the Apache HTTPD instance, Apache Portable Runtime (APR) and Perl-Compatible Regular Expressions Library (PCRE) are downloaded on the server (Apache Software Foundation 2015). The first one is used to both create and maintain software libraries for Application Programming Interfaces (APIs), to which software developers can code without the need to modify or customize the code depending on the platform (Apache Portable Runtime Project 2015). PCRE on the other hand provides a library for Perl 5, a general-purpose code language, which can complete a wide range of tasks from system administration to network and web development (Robert 2015).

Simply put, APIs such as APR and PCRE provide an interface between two systems that are possibly written in different code languages and thus making development processes more straightforward. After acquiring both of the aforementioned, as well as downloading OpenSSL, the Apache HTTPD server can be compiled by running the following syntax inside *bin* directory:

./configure --enable-ssl --enable-proxy-http --enable-proxy-connect --enable-headers --enable-rewrite --prefix=/<location of the Apache HTTPD installation> --with-ssl=/<location of the OpenSSL download> --with-included-apr --with-pcre

The aforementioned compiling statement adds SSL, HTTP Proxy and Proxy connect, and headers to the *httpd.conf* file. These are used to configure the server appropriately, but the configurations can also be added after the server has been compiled. Separately downloaded APR utilities must be located in the HTTPD server root inside *scrlib* directory, thus the use of - *with-included-apr* (Apache Software Foundation. 2015). Sometimes the system might not find PCRE installation, so the script must be pointed to use the *-with-pcre* parameter (Apache Software Foundation 2015).

3.1.2 httpd.conf

The Apache HTTPD main configurations are done to the *httpd.conf* file. The default configuration file contains multiple commented-out configurations to implement different elements to the frontend server. Prior to other setups however, the server should be bind to a port (Hock-Juan 2015). Apache uses port 80 as a default port to transmit data between the browser and server using TCP protocol (Gite 2005), but in theory the listen port can be configured to listen to any open port.

To be able to access the resource from the internet browser, the *httpd.conf* file needs a *ServerName* parameter. Either the server's hostname or IP-address is addressed in the configuration file (Hock-Juan 2015), in order to make the web content accessible from the browser. The configuration file also holds the location of the root directory for the web documents, which is a location on the server, which holds for example HTML-files.

One of the most fundamental parts of this thesis is the login handler, which is at the bottom of the configuration file. The handler is used to direct the user to an appropriate page after successful or unsuccessful login attempt. The login form on the frontend Apache server is dedicated to transfer user credentials to Shibboleth IdP, which processes the credentials and relays them securely to external applications. A more elaborate description of this process is explained in the later chapters.

```
<Location "/idp">
    AuthFormProvider file
    ErrorDocument 401 "/index.html"
    AuthUserFile "/usr/oppari/conf/.passwords"
    AuthType form
    AuthName realm
    Session On
    SessionCookieName session path=/
    Require valid-user
    ProxyPass ajp://localhost:8019/idp
    ProxyPassReverse ajp://localhost:8019/idp
</Location>
```

Different functionalities and features can be added to the Apache HTTP server by taking advantage of its modularity with the use of LoadModules (Gite 2006). The login form that collects user's credentials uses the *auth_form_module*, which allows the usage of a login form without the need to manually input excess code. The loaded modules are declared at the top of the configuration file above and the listing consists of other modules in addition to *auth_form_module*, which are all used to either pass the traffic between Apache HTTP and Shibboleth IdP or to enhance the security of the frontend server.

To prevent credentials from showing as plain text, it is vital to implement the *mod_session_crypto* module, which encrypts the messages before they are sent to external locations or stored in local databases (Apache Software Foundation 2015). This could potentially expose the site for cross scripting attacks, because contents of the user session can be interpreted from HTTP headers and used to get information on the underlying system (Apache Software Foundation 2015).

The users in this case are stored locally on the HTTPD server, as can be seen from the *AuthUserFile* attribute. This is not a good practice, as it is not very secure or effective, but for this proof of concept version the employer recommended this solution. Typically a directory, such as Microsoft's Active Directory, should be used. Active Directory is a database that can handle a large number of searches and read tasks, but less changes and updates (Microsoft 2015). For the aforementioned reasons, it functions well as a user directory, as user information has to be altered rarely.

The whole *httpd.conf* file can be found in the appendices. It includes some of the default comments to explain the purpose of specific elements.

3.2 Apache Tomcat

The previous section briefly disclosed the term backend server. In comparison to the frontend server, a backend is not per se visible to the user. The most typical usages for backend servers are handling the server's side, applications and connecting to a database or a directory (Girdley 2014). To put it plainly, a backend server does the operative and computing tasks that frontend server does not handle. A backend server can communicate directly with the connected frontend, or frontends, or intermediate programs can be used to pass messages between the two interfaces.

As deducted from the depiction above, the term backend server is rather vague, as it can mean anything that resides behind the frontend servers. Apache Tomcat is commonly referred to as an application server, as it is capable of deploying Java applications. Choosing an application server usually needs to be decided upon the application's requirements, such as whether the applications need to be fully compatible with certain Java features or if it needs Enterprise JavaBean capability, which is a feature that makes application development easier (Campbell 2007).

Shibboleth can be deployed on Apache Tomcat, and therefore it was chosen for this thesis as a counterpart for Apache's frontend server. It is also more light-

weighted in comparison with other options, and it doesn't require much space, which makes it ideal for a test environment (Campbell 2007). Other option could have been JBoss, which is also an open source application server, somewhat similar to Tomcat.

3.2.1 Java

Java is a programming language, typically used to create web content and applications (Beal 2015). This widespread programming language is also the foundation of Shibboleth IdP.

Java is not included in the Apache Tomcat download, so it needs to be downloaded separately from their website. Once downloaded and installed in the same server as the Tomcat, the Tomcat *catalina.sh* file needs to point to the exact location of the Java installation. The aforementioned script is used to start the Tomcat server.

3.2.2 AJP Connector

Apache Jserv Protocol (AJP) Connector is a component that communicates with a web connector using AJP protocol, to adapt SSL processing between the frontend and backend servers, to ensure security when passing user-related credentials (Apache Tomcat 2015). Protocols such as HTTP and HTTPS are other plausible options, but AJP protocol provides faster processing, as it requires less bandwidth compared to the aforementioned protocols and is supported by the Apache installations (Well House Consultants 2008). Because of its easy implementation in Apache Tomcat and Apache HTTPD, AJP is used in this thesis to provide a means of communication between the two servers.

Below is the configuration for the AJP Connector, a snippet from the *server.xml* file. Frontend server's AJP connector is defined in *httpd.conf* file, which was discussed in the previous chapter.

<!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Define a non-SSL HTTP/1.1 Connector on port 8080 ->

<!-- A "Connector" using the shared thread pool-->

<!-- Define a SSL HTTP/1.1 Connector on port 8443

This connector uses the JSSE configuration, when using APR, the

connector should be using the OpenSSL style configuration

described in the APR documentation -->

<!--

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="/usr/oppari/idpcerts/idpself.keystore"

keystorePass="<insert keystore password here>" />

-->

<!-- Define an AJP 1.3 Connector on port 8009 -->

<Connector port="8019" protocol="AJP/1.3" redirectPort="38443" address="localhost" scheme="https"

proxyPort="443" tomcatAuthentication="false" />

AJP connector is connected to port 8009, an open TCP port, as a medium between the backend and frontend server communication (NTU 2015). Technically any open port can be chosen to be the connector. The *tomcatAuthentication* parameter is set to false, as Shibboleth IdP is used to make the authentication decisions instead of the Tomcat (Apache Tomcat 2015).

3.3 OpenSSL

OpenSSL is used to implement TLS and SSL protocols, as well as a cryptographic library to provide a secure interface to collect user credentials (OpenSSL Project 2015). OpenSSL allows the encryption of the data flow, to

prevent plain text data transmissions (SSLShopper 2010) that could be easily caught and interpreted.

4 SHIBBOLETH IDENTITY PROVIDER

Shibboleth IdP is open source software used to hold Digital Identity information and assert it to a Service Provider (SP). IdP handles authentication requests from SPs and can provide a variety of user related identifiers to allow more sophisticated authentication process. This for one means better user experience as certain user information can be passed on to the service, reducing the need of manual data input (Shibboleth 2015). These identifiers are referred to as attributes when they are nested inside a SAML assertion (OASIS 2008, 8).

Shibboleth is a good choice for IAM environments, because of it is open source and free of charge. It means that the product is easily customisable and great amount of support is available in Shibboleth's forums and free articles. One of the biggest rivals to Shibboleth has been Microsoft's Active Directory Federation Services (ADFS), which operates only on paid licensed Windows platforms (Gasper 2014), making it less favorable for this use case.

4.1 How Shibboleth Works

Shibboleth software products work as a separate platform for user identifying purposes. Authentication decisions are made by sending minimal identity information necessary between Shibboleth's two major components: IdP and SP (Shibboleth 2015). In this thesis, the identifying information a user possesses, are the user's credentials, collected by the login handler configured at the Apache HTTPD server.

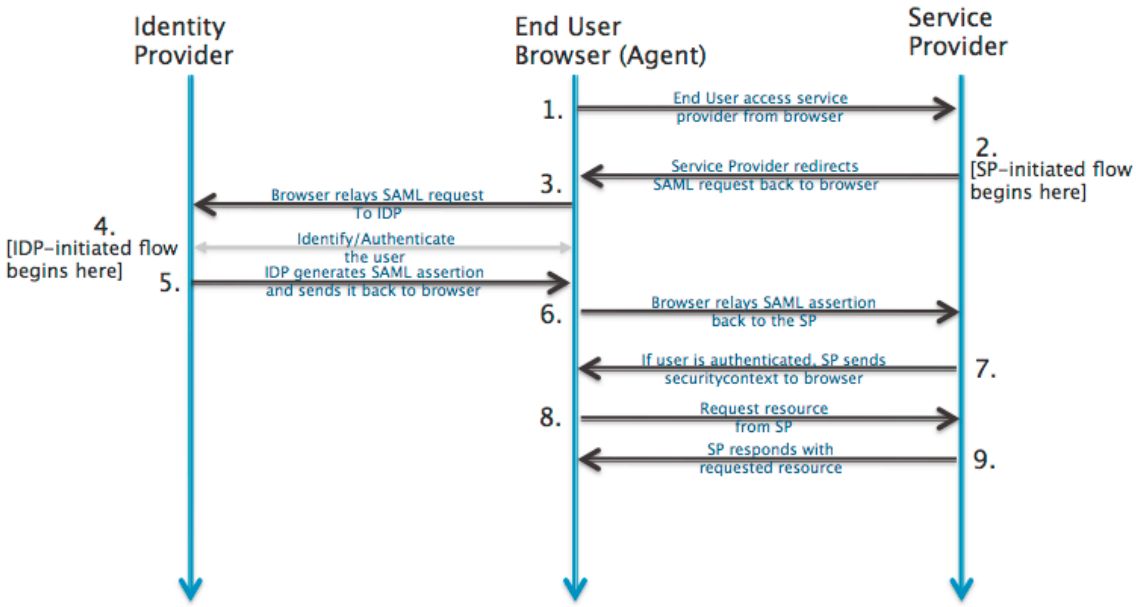
Even though only the most minimal amount of information is sent between the providers, the authentication or authorization decisions can be highly sophisticated, depending on the amount of attributes available. Attributes refer to identifying information upon which authentication and authorization decisions can be made, for example, dividing users to employee and employer attributes. If user has the employee attribute, he or she may be allowed to access different resources than the users with the employer attribute.

4.1.1 Identity Provider and Service Provider

IdP and SP collaborate so that IdP hands over information of users to SP (Shibboleth 2015). Identity federation, or a trust network, is established upon these two components, where IdP holds the digital identities of all users and SP keeps the resources safe.

A typical usage could be described as follows: a user arrives at a website and tries to access a resource or an application, through a web browser. They are forced to authenticate at IdP, in other words, their user credentials can be collected through a login form supported by the IdP. After the user is authenticated accordingly by the information IdP supplies the SP, the user can access the protected resource (Shibboleth 2015).

Authentication request can be initiated either by IdP or SP. The web browser is the information broker used to forward requests and responses between IdP and SP (Okta 2015). A high-level depiction of the flow of information is displayed below.



Picture 1. High-level demonstration of IdP and SP collaboration (Okta 2015).

Shibboleth's trust establishment between the IdP and SP is achieved using by exchanging information between the two in the form of metadata (Cantor 2015). The requests and responses are relayed between the trusted parties using the SAML 2.0 standard, which is based on another standard called Extensible Markup Language (XML). SAML is described in detail below, but in short, it is a format for exchanging information between programs or computers, even across networks (W3 2015).

4.1.2 Metadata and Certificates

To determine the parties involved in the identity federation system, the IdP and SP must identify itself and one another. Shibboleth uses metadata to pass on identifying information of the provider, to prove it is who it claims to be. When the configurations in the form of signed metadata are sent from the IdP to SP, or the other way around, the metadata is consumed. After the information has been exchanged, the providers are added to the list of trusted partners and the receiving party receives information on how to securely communicate with the other provider. (Klingenstein 2010)

The certificate signing can be self-signed or signed by a certificate authority (CA)(Krienke 2015), which means who determines the validity of the sending or receiving party. As the certificate between the providers is not used on the browser facing connections (Krienke 2015), a self-signed certificate is sufficient. A certificate signed by a CA is commonly required when customers need to be informed that the server's information is backed up and verified by a trusted source, such as Verisure (Kyrnin 2007).

Shibboleth foundation recommends the certificate to be placed inside the provider's metadata (Shibboleth 2015). An example of the metadata is disclosed in the appendix. If the transmission of dataflow should be encrypted, Public-key cryptography can be used by placing a separate certificate for signing inside the metadata, in addition to the certificate used to identify the provider (Krienke 2015).

4.1.3 SAML 2.0 Assertion

SAML is a XML-based standard that is developed and maintained by OASIS Security Services Technical Committee (SSTC) (OASIS 2008, 8). SAML has several use cases, as apart from describing and transferring digital identity information, it can also be used to fuel single sign-on and other web services (OASIS 2008, 8). Regarding this thesis, SAML is used to declare protocol syntax and structure of information exchange. As collaborating IdP and SP communicate via SAML assertions, they can have a mutual understand how user's digital identities are created and communicated (OASIS 2008, 8).

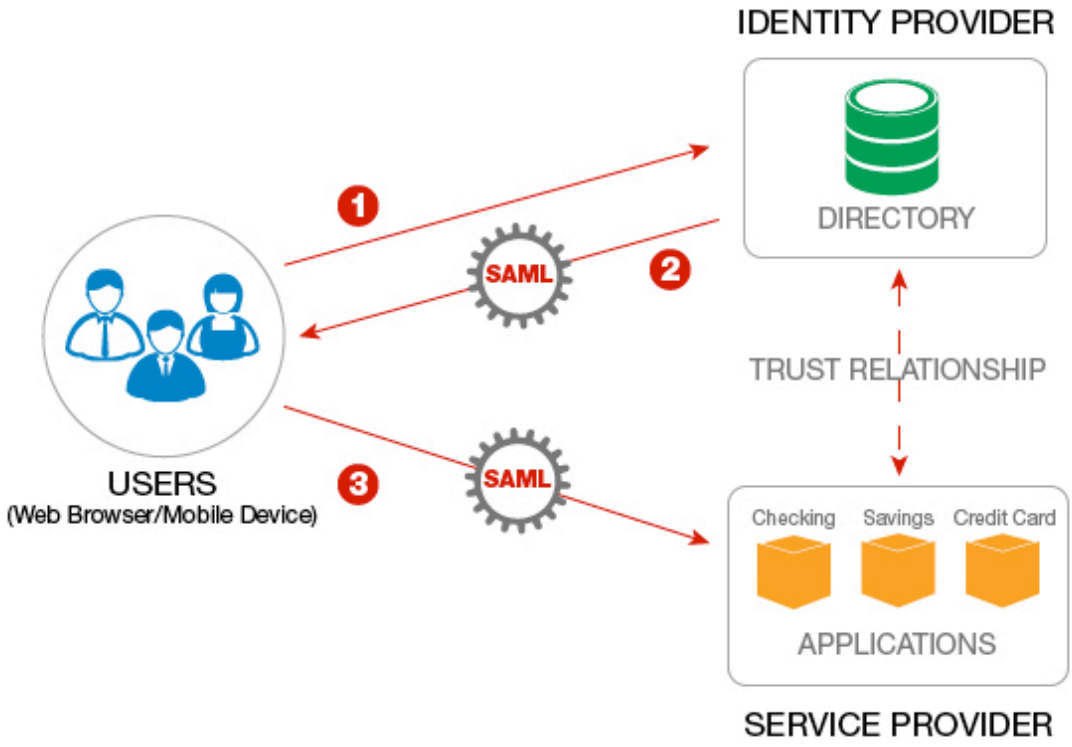
SAML assertions can carry three types of messages: authentication, attributes and authorization decision. Firstly it can authenticate a user and give it access to specific resources, which is usually done by the IdP. Secondly the assertion holds information on what attributes the user holds, such as a title or access right level. Lastly the assertion may contain authorization decision, which tells the SP whether the user has or has not been granted access by the IdP. (SAML 2007)

The previously described messages can be in the form of requests or responses. SP forms the requests, when they request information on a user, and response assertions are sent by the IdP (Oracle 2015). The response contains information of the user that is forwarded to the SP via the user’s browser (Pisarkiewicz 2014).

The SAML assertions are bind to a Hypertext Transfer Protocol (HTTP) POST method. This allows the IdP to interact with the user and relay messages to the SP using the user’s web browser. (OASIS 2008, 21-22) HTTP is an underlying protocol of all internet content, which determines how messages should be formatted and transmitted, as well as how web browsers and servers should response to different commands (Beal 2015).

In other words, if SAML assertions were not bind to HTTP messages, the IdP could not get user related information, and the information could not be sent to

the SP. The key element is the web browser, which serves as a medium between the components. In this kind of use case, IdP and SP have no direct connection with each other.



Picture 2. SAML assertion from the IdP to SP (Pisarkiewicz 2014).

As can be seen from the picture above, the user's web browser, or some other internet facing interface, sends a typical HTTP request to the IdP. This response may contain input from the web browser, such as user's credentials. The SAML assertion is placed inside the HTTP response, and is forwarded to the SP, which reads the SAML assertion. (Pisarkiewicz 2014) The process above is IdP initiated authentication. If the authentication was requested by the SP, the SAML request assertion would be first generated by the SP and relayed to the IdP.

There are several other standards that are used to provide identity federation features, albeit a little differently. One option is OAuth that can be used by SPs, to import contact information, commonly from email providers (Mehta). For ex-

ample LinkedIn allows users to fetch information from contacts from email provider Gmail.

Another example is OpenID, which bears more resemblance to SAML, but instead of being used by enterprises, they are more commonly deployed to consumer applications or services (Mehta). OpenID is widely implemented by several large-scale internet applications, but due to its lack of uniform in design, it has become harder to implement and some claim it has lost its appeal. Facebook Connect has also partly taken over OpenID's market sector by allowing third party applications to fetch user data from Facebook. (Gilbertson 2015) However, big internet organisations, such as Google, still use OpenID (Google 2015).

4.1.4 Debugging Assertions

IdP installation should be debugged whether it can handle requests and send responses, and for this thesis the configurations are tested against a website called TestShib (www.testshib.org). TestShib requires the IdP's metadata to be inputted manually to the website, in other words, the *idp-metadata.xml* file is copied from the server and enumerated to TestShib via browser. Respectively, TestShib's information needs to be added to *metadata-providers.xml* configuration file on the server. Below is an example of an authentication requests sent by the SP to the IdP.

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" AssertionConsumerServ-
iceURL="https://sp.testshib.org/Shibbleth.sso/SAML2/POST"

Destination=https://lkoppari.office.trusteq.com/idp/profile/SAML2/Redirect/SSO
ID="_3a4f4ff9b9e26f13712619ab4c83fe16"

IssueInstant="2015-11-27T10:05:26Z"

ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Version="2.0">

<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://sp.testshib.org/shibboleth-sp</saml:Issuer>

<samlp:NameIDPolicy AllowCreate="1" />

</samlp:AuthnRequest>

The request is received and handled by the IdP, which can send authentication and authorization statements as a response to the requesting SP (Oracle 2015). The response sent by the IdP built in this thesis is lengthy and thus is enclosed in the appendices. The response contains information from the IdP that it has successfully authenticated the user.

The response contains a certificate, to prove the message originates from an IdP that the SP trusts. It declares a set of algorithms and standards that govern the information exchange. If the user inputs wrong or invalid credentials, the response would tell the SP that the authentication failed and the user is not allowed to access SP's resources.

The above could also be summarised as follows: the user tries to access a resource, in other words the SP, through a web browser. The SP, however, does not know the user's identity and therefore cannot confirm whether the user is allowed to access the resources. Thus the user is redirected to a login page provided by the IdP. Upon inputting right credentials, the user is given an identity the SP can interpret and understand.

5 CONCLUSION

Complex information systems cause a lot of administrative overhead and security issues by default. However, as expressed through countless of studies on information security, the weakest link most typically turns out to be the user. Different passwords policies can be deployed to somewhat reduce the issues with weak passwords, but policies itself cannot prevent for example using the same password in multiple systems.

Thus in terms of security, Identity Federations make it easier to manage user-related information, such as usernames and passwords. When this information can be delivered to other independent systems, the need to store user credentials at each access point is reduced, which make it easier for users to access the resources they need. In addition, users no longer need to remember many different passwords, but can rely on one strong and complex password, which enhances information security a great deal.

Assumedly cutting the costs of holding extensive user repositories, large-scale businesses are undoubtedly interested in creating a more secure information system, to combat possible threats and complications that poorly constructed information system may pose. On the other hand, some businesses might be interested in creating more collaborative environment with its partners or customers, to perform better as a company.

This thesis provides a simple example of an environment, which can be used to achieve the aforementioned. As demonstrated above, the environment does not need expensive licenses or platforms, as everything from the servers to the Shibboleth product family is free of charge and open source. Actually, many of the free and open source options, such as Apache's HTTPD and Tomcat, are regarded as better than their commercial counterparts (Campbell 2007).

The environment still needs some future enhancements, such as a proper user directory to enumerate user credentials, instead of storing them locally at Apache HTTPD server. As of now the environment servers its very fundamental

purpose: it gathers user-inputted information and forwards them to Shibboleth IdP. In real life scenarios, the environment would be complex also because they are typically communicating with several IdPs or SPs. The Shibboleth product family can also be used as an addition to other IAM software, which expand the capabilities of user management even further.

As witnessed in this thesis, it is very hard to build this kind of an environment without prior knowledge on the subject. However, there are plenty of articles on the internet that provided helpful while composing this thesis. Some solutions needed a lot more groundwork than others, but as a concluding statement, writing this thesis took a lot less time than managing to deploy a fully-functioning environment using the components I described above.

REFERENCES

Apache Portable Runtime Project 2015. Welcome. Cited: 14.10.2015. https://apr.apache.org.

Apache Software Foundation 2015. Apache Module mod_session_crypto. Cited: 25.11.2015. http://www.apache.org > HTTP Server > Documentation > Version 2.5 > Modules

Apache Software Foundation 2015. Compiling and Installing. Cited: 14.10.2015. https://httpd.apache.org > HTTP Server > Documentation > Version 2.4.

Apache Tomcat 2015. The AJP Connector. Cited: 15.10.2015. https://tomcat.apache.org > Tomcat 7.0 > Connectors.

Apache Tomcat. 2015. The AJP Connector. Cited: 26.11.2015. https://tomcat.apache.org > Tomcat 7.0 > Configuration

Beal, V. 2015. HTTP – HyperText Transfer Protocol. Cited: 15.12.2015. http://www.webopedia.com > TERM > H.

Beal, V. 2015. Java. Cited: 9.12.2015. http://www.webopedia.com > TERM > J.

Campbell, J. 2007. Jboss, Geronimo or Tomcat? Cited: 9.12.2015. http://www.javaworld.com > Enterprise Java > Open Source.

Cantor, S. 2015. Metadata. Cited: 20.11.2015. https://wiki.shibboleth.net > Shibboleth Concepts > Metadata.

EDUCAUSE 2015. 7 Things you should know about... Federated Identity Management.

Finnish Ministry of Finance 1999. Tietoturvallisuus ja lainsäädäntö. 523/1999.

Gaedke, M., et al. 2015. A Modelling Approach to Federated Identity and Access Management.

Gasper, J. 2014. Why Shibboleth is a Great Alternative to Active Directory Federation Service. Cited: 9.12.2015. https://www.unicon.net > about > articles.

Gilbertson, S. 2011. OpenID: The Web's Most Successful Failure. Cited: 15.12.2015. http://www.webmonkey.com > Identity

Girdley, M. 2014. Front-end vs Back-end. Cited: 20.9.2015. http://codeup.com > Blog > Search: "Back-end".

Gite, V. 2005. Linux Iptables: HowTo Block or Open HTTP/Web Service Port 80 & 443. Cited: 28.9.2015. http://www.cyberciti.biz > Howtos and Tutorials > Search "Linux Iptables:".

Gite, V. 2006. Howto: Apache adding new modules. Cited: 29.9.2015. http://www.cyberciti.biz > Howtos and Tutorials > Search: "Howto: Apache adding".

Google 2015. Migrating from OpenID 2.0 to OpenID Connect. Cited: 15.12.2015. https://developers.google.com > Products > Google Identity Platform

Hock-Chuan, C. 2015. How to Install and Get Started with Apache 2. Cited: 28.9.2015. http://www3.ntu.edu.sg/home/ehchua/programming > Apache 2.2 – How to Install.

Hudson, S. 2011. Identity and Access Management: The Foundation for Secure, Efficient, and Compliant Enterprise Application Environments. Cited: 11.10.2015.

IBM 2015. IBM HTTP Server. Cited: 9.12.2015. http://www-03.ibm.com > IBM Software > Products > Application infrastructure > Application optimization.

Josang, A., et al. 2015. Trust Requirements in Identity Management.

Klingenstein, N. 2010. BuildAFederation. Cited: 11.11.2015. https://wiki.shibboleth.net > Shibboleth 2 > Installation and Configuration > Configuration.

Krienke, J. 2015. T. X.509 Certificates in Federation Metadata. Cited: 11.11.2015. https://spaces.internet2.edu > InC-Federation > InCommon Federation > Metadata Administration.

Kyrnin, J. 2015. Signed vs. Self-signed Certificates. Cited: 22.11.2015. http://about.com > About Tech > Web Design & HTML > Web Server Management > Web Security > Secure Sockets Layer – SSL.

Mehta, L. 2014. SAML, OAuth, OpenID. Cited: 21.11.2015. http://resources.infosecinstitute.com > SAML, Oauth, OpenID.

Microsoft 2015. So What Is Active Directory? Cited: 9.12.2015. https://msdn.microsoft.com > Windows desktop applications > Develop Desktop technologies > Security and Identity Directory, Identity, and Access Services > Directory Services > Directory Access Technologies > Active Directory Service Interfaces > Active Directory Service Interfaces Quick-start Tutorials > Accessing Active Directory Using Visual Basic.

NTU 2015. How to Connect Tomcat 6 to Apache HTTP Server 2. Cited 26.11.2015. https://www3.ntu.edu.sg/home/ehchua/programming/index.html > Apache 2 with Tomcat

Okta 2015. Single Sign-on with Okta. Cited: 26.11.2015. http://developer.okta.com > Docs > Guides > Planning for SAML.

OpenSSL Project 2015. OpenSSL. Welcome to the OpenSSL Project. Cited: 9.11.2015. https://www.openssl.org.

Pisarkiewicz, C. 2014. How SAML is Used for Single Sign-on (SSO). Cited: 15.12.2015. http://www.forumsys.com > Blog > Categories > SAML.

Robert, K. 2015. What is Perl? Cited 14.10.2015. http://perldoc.perl.org > Overview > perlintro.

SAML.xml.org 2007. Assertions. Cited: 23.11.2015. http://saml.xml.org > Wiki Knowledgebase > Assertions.

Shibboleth 2015. Shibboleth Identity Provider. Cited: 3.9.2015. https://shibboleth.net > Products products/identity-provider.html.

Shibboleth 2015. What's Shibboleth? Cited: 22.10.2015. https://shibboleth.net > About.

SSLShopper 2010. How to Create and Install an Apache Self Signed Certificate. Cited: 9.11.2015. https://www.sslshopper.com > SSL News.

Tuomi, A. 2015. Haka perustuu luottamusverkostoon. Cited: 3.9.2015. https://confluence.csc.fi > Haka-käyttäjätunnistusjärjestelmä > Luottamusverkko.

W3C 2015. What is XML? Cited: 20.11.2015. http://www.w3.org > Standards >XML Technology > XML Essentials.

Wallen, J. 2009. Techrepublic. 10 Things You Should Do to Secure Apache. Cited: 20.9.2015. http://www.techrepublic.com > Blogs > 10 Things.

Well House Consultants 2008. http, https and ajp - comparison and choice. Cited: 15.10.2015. http://www.wellho.net > Apache Http Server & Tomcat > deploying Apache httpd and Tomcat > module A655 > 1549 http, https and ajp – comparison and choice.

WhatIs 2015. Definition: Front-end. Cited: 20.9.2015. http://whatis.techtarget.com > Browse Definitions by Alphabet > F.

Appendix: httpd.conf file

#Listen port: this is the IP address and port the website can be accessed via browser.

Listen 192.168.101.166:80

#Modules: modules that add to the functionality of Apache HTTPD server

LoadModule authn_core_module modules/mod_authn_core.so

LoadModule authz_host_module modules/mod_authz_host.so

LoadModule authz_groupfile_module modules/mod_authz_groupfile.so

LoadModule authz_user_module modules/mod_authz_user.so

LoadModule authz_core_module modules/mod_authz_core.so

LoadModule access_compat_module modules/mod_access_compat.so

LoadModule auth_basic_module modules/mod_auth_basic.so

LoadModule auth_form_module modules/mod_auth_form.so

LoadModule socache_shmcb_module modules/mod_socache_shmcb.so

LoadModule reqtimeout_module modules/mod_reqtimeout.so

LoadModule request_module modules/mod_request.so

LoadModule session_module modules/mod_session.so

LoadModule session_cookie_module modules/mod_session_crypto.so

LoadModule filter_module modules/mod_filter.so

LoadModule mime_module modules/mod_mime.so

LoadModule log_config_module modules/mod_log_config.so

LoadModule env_module modules/mod_env.so

```
LoadModule headers_module modules/mod_headers.so

LoadModule setenvif_module modules/mod_setenvif.so

LoadModule version_module modules/mod_version.so

LoadModule proxy_module modules/mod_proxy.so

LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

LoadModule session_module modules/mod_session.so

LoadModule session_cookie_module modules/mod_session_cookie.so

LoadModule ssl_module modules/mod_ssl.so

LoadModule unixd_module modules/mod_unixd.so

LoadModule status_module modules/mod_status.so

<IfModule unixd_module>

User laura

Group laura

</IfModule>

ServerName lkoppari.office.trusteq.com

<Directory />

Options FollowSymLinks

AllowOverride none

Order deny,allow

Deny from all

</Directory>

DocumentRoot "/usr/oppari/htdocs"
```

<Directory "/usr/oppari/htdocs">

Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory> <IfModule dir_module>

DirectoryIndex index.html

</IfModule>

The following lines prevent .htaccess and .htpasswd files from being

viewed by Web clients.

<Files ".ht*">

Require all denied

</Files>

ErrorLog "logs/error_log"

LogLevel warn <IfModule log_config_module>

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

LogFormat "%h %l %u %t \"%r\" %>s %b" common

</IfModule>

The location and format of the access logfile (Common Logfile Format).

CustomLog "logs/access_log" common

</IfModule>

```
<IfModule alias_module>
# ScriptAlias: This controls which directories contain server scripts.
ScriptAlias /cgi-bin/ "/usr/oppari/cgi-bin/"
</IfModule>
<Directory "/usr/oppari/cgi-bin">
AllowOverride None
Options None
Require all granted
</Directory>
<IfModule mime_module>
# TypesConfig points to the file containing the list of mappings from
# filename extension to MIME-type.
TypesConfig conf/mime.types
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
</IfModule>
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
<Location "/idp">
    AuthFormProvider file
```

```
    ErrorDocument 401 "/index.html"
    AuthUserFile "/usr/oppari/conf/.passwords"
    AuthType form
    AuthName realm
    Session On
    SessionCookieName session path=/
    Require valid-user
    ProxyPass ajp://localhost:8019/idp
    ProxyPassReverse ajp://localhost:8019/idp
</Location>
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
```

Appendix: Metadata and Certificate

<?xml version="1.0" encoding="UTF-8"?>

<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:shibmd="urn:mace:shibboleth:metadata:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" entityID="https://lkoppari.office.trusteq.com/idp/shibboleth">

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0 urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol">

<Extensions>

<shibmd:Scope regexp="false">trusteq.com</shibmd:Scope>

</Extensions>

<KeyDescriptor use="signing">

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIDVzCCAj+gAwIBAgIUA3oeeV3blNqhPQ2YaK7biik8AVMwDQYJKoZIhvcNAQEF

BQAwJjEkMCIGA1UEAwwbbGtvcHBhcmkub2ZmaWNlLnRydXN0ZXEuY29tMB4XDTE1

MDcyNDA3MDc1NFoXDTM1MDcyNDA3MDc1NFowJjEkMCIGA1UEAwwbbGtvcHBhcmku

b2ZmaWNlLnRydXN0ZXEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMII
BCgKC

AQEAtwG7AiCEhSD2FoLb0ZDijiKCkJvLFXW7cOMMGfzLHDoux5RpX4KPVpl
b4Ngr

7zhehs0ccYcoh3g3q4c456HRM67MgdOtI3DPI+sCxL4lJRCPqtC3dl65VJnkY6B
j

eCw7KQHoDtuXT9HEM5ir/uuX8+is1ZkRJH1EEBNNHW13Bmk0c6Wp37sO7T
K0qpl5

w83kzI4AKfwW5BKrdNvPAzVdjOijDxsUj5u2fH/52QicjYC5+xe2gb0QanX9fM2U

kFehZXnrm+oyavds97rklUkA7Lgj15k0Aqpbvs2C/29pdyKu0kfTuEywm3+h9SUv

AUJyX2OB8KMQQxW+skv0dfcdEQIDAQABo30wezAdBgNVHQ4EFgQUi7E9w
sWY8luR

Dz6bhP3cX8JBG3QwWgYDVR0RBFMwUYIbbGtvcHBhcmkub2ZmaWNlLnRyd
XN0ZXEu

Y29thjJodHRwczovL2xrb3BwYXJpLm9mZmljZS50cnVzdGVxLmNvbS9pZHAvc
2hp

YmJvbGV0aDANBgkqhkiG9w0BAQUFAAOCAQEAjnn1nbl2cRQBF47BjznFlNO
NSwth

Dwz6Mx8NA0LT4knQyDVDw3BmOz/yGZaUrShQPVFsetq5xcXXahvEghHtLYY
BB9nO

ARXrbFihlJ6Wi80oa6AnVD1YCOpeAcMEpCazMP9ebCoT0sYMues9++UkxdG
G441L

OS0PlE8HjUXXSbpd/CtD9RQMGJcaNSECQmHep2LX8k53gzLf2unSaOvBVZl
22scK

W5l1GCaLvwPave4ZVWYMPRyVuaHP5DFHXo20NcKKOF35X0En7GxrlitzOO
+o1wgN

31A7LzUP2sm6OXmDULA3i33cKWSmxnDtXXU03GAACo9MJkpYWrCSjOqdx
Q==

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

Appendix: SAML Response


```
<saml2p:Response
Destination="https://sp.testshib.org/Shibboleth.sso/SAML2/POST"
              ID="_d7be5ef906660a2c8b465e6e60eed3d3"
              InResponseTo="_a1a83cbe33543bbecd285286beea3ece"
              IssueInstant="2015-11-30T08:00:03.077Z"
              Version="2.0"
              xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">
   <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity"
               xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
               >https://lkoppari.office.trusteq.com/idp/shibboleth2</saml2:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" />
         <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
         <ds:Reference URI="#_d7be5ef906660a2c8b465e6e60eed3d3">
            <ds:Transforms>
               <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
               <ds:Transform       Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
            </ds:Transforms>
            <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

<ds:DigestValue>MWJXP5R5/VjS9N6fai/pCqSsLBw=</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
lVGCfmMAmKBPQJXe4Ir7PCBtc9yZeX7t4xbLs17PTbTz++AGvO4zOjgQaPT4
6sKaouXDytXA2XMsaqQm+BdcY4KmCEc4CMa1gPFXB6SDqoL5tFk1XNNWjh
VCM6OTLBedW/VJieP2eUAlk1kzNOx1pr1RbP7cN7LTzAC5cVCSegKlrGkNmx
zmDLDKX8BMSZ6qXfTphkfDNQhRgLRtkAjRl8Y/G26sgrOgaH7L6aflIUfvm/l4H
HffKAvZTwS-
RHjD2RU20tvER0+6quggYoLsUT7jFfAcJY2ga+K3z64ryMZxzqyg5XyqMBDLK/
2Z2u7CQcOM4Py5Nzq+9TxQvFV6g==</ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>

<ds:X509Certificate>MIIDVzCCAj+gAwIBAgIUA3oeeV3blNqhPQ2YaK7biik8AV
MwDQYJKoZIhvcNAQEFBQAwJjEkMCI-
GA1UEAwwbbGtvcHBhcmkub2ZmaWNlLnRydXN0ZXEuY29tMB4XDTE1MDcy
NDA3MDc1NFoXDTM1MDcyNDA3MDc1NFowJjEkMCIGA1UEAwwbbGtvcHBh
cmkub2ZmaWNlLnRydXN0ZXEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8
```

AMIIB-
CgKCAQEAtwG7AiCEhSD2FoLb0ZDijiKCkJvLFXW7cOMMGfzLHDoux5RpX4
KPVplb4Ngr7zhehs0ccYcoh3g3q4c456HRM67MgdOtI3DPI+sCxL4lJRCPqtC3d
l65VJnkY6BjeCw7KQHoDtuXT9HEM5ir/uuX8+is1ZkRJH1EEBNNHW13Bmk0c
6Wp37sO7TK0qpl5w83kzI4AKfwW5BKrdNvPAzVdjOijDxsUj5u2fH/52QicjYC5+
xe2gb0QanX9fM2UkFehZXnrm+oyavds97rklUkA7Lgj15k0Aqpbvs2C/29pdyKu0
kfTuEywm3+h9SUvAUJyX2OB8KMQQxW+skv0dfcdEQIDAQABo30wezAdBgN
VHQ4EFgQUi7E9wsWY8luRDz6bhP3cX8JBG3QwWgYDVR0RBFMwUYIbbGt
vcHBhcmkub2ZmaWNlLnRydXN0ZXEuY29thjJodHRwczovL2xrb3BwYXJpLm9
mZmljZS50cnVzdGVxLmNvbS9pZHAvc2hpYmJvbGV0aDANBgkqhkiG9w0BA
QUFAAOCAQEAjnn1nbl2cRQBF47BjznFlNONSwthDwz6
Mx8NA0LT4knQyDVDw3BmOz/yGZaUrShQPVFsetq5xcXXahvEghHLYYBB9n
OARXrbFihlJ6Wi80oa6AnVD1YCOpeAcMEpCazMP9ebCoT0sYMues9++Ukxd
GG441LOS0PlE8HjUXXSbpd/CtD9RQMGJcaNSECQmHep2LX8k53gzLf2unSa
OvBVZl22scKW5l1GCaLvwPave4ZVWYMPRyVuaHP5DFHXo20NcKKOF35X0
En7GxrlitzOO+o1wgN31A7LzUP2sm6OXmDULA3i33cKWSmxnDtXXU03GAA
Co9MJkpYWrCSjOqdxQ==</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
   <saml2p:Status>
      <saml2p:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
   </saml2p:Status>
   <saml2:Assertion ID="_906c4fa0acc5c923efc6e7a88ad69ea9"
                    IssueInstant="2015-11-30T08:00:03.077Z"
                    Version="2.0"
                    xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml2:Issuer   Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://lkoppari.office.trusteq.com/idp/shibboleth2</saml2:Issuer>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <ds:SignedInfo>
            <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <ds:Reference URI="#_906c4fa0acc5c923efc6e7a88ad69ea9">
               <ds:Transforms>
                  <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                  <ds:Transform   Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
               </ds:Transforms>
               <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />

<ds:DigestValue>bWLO0e/rChnqGT5WyLvGHf5cbO4=</ds:DigestValue>
            </ds:Reference>

```
        </ds:SignedInfo>

<ds:SignatureValue>iQApg7nBVO+/GhdN3kHK/QdJG4ZQ5hNYDTEZbWM354
JK7JvWYZizXrpmbgF3I+x3YGgeW8yTMDEnnWj7662vvlbKYA6LIPE6TmSkJC
9cyqf1f5ZO7ScjIwfpYSUJHRdkAQF6h4Yu+q0nlo4KZrEC07c3BwVgq69QT5D8
kaGW82zXAffoZ70WCem6SRFY6/hld9PJmonwfP9S8cqV3K1b4D7gSAdrI256i
sIKD1C4JQdSyTUwWgzOLjCuZ8AiXBG/Xmx7HN2SfQgvvVuIWLTgysII901Wx
NJnyM/TkMoU7imX3dfsh5hGqoVFZ9TI2rHVDAzczcINT3xkoofOfaVsPw==</ds:
SignatureValue>
        <ds:KeyInfo>
          <ds:X509Data>

<ds:X509Certificate>MIIDVzCCAj+gAwIBAgIUA3oeeV3blNqhPQ2YaK7biik8AV
MwDQYJKoZIhvcNAQEFBQAwJjEkMCI-
GA1UEAwwbbGtvcHBhcmkub2ZmaWlLnRydXN0ZXEuY29tMB4XDTE1MDcyN
DA3MDc1NFoXDTM1MDcyNDA3MDc1NFowJjEkMCIGA1UEAwwbbGtvcHBhc
mkub2ZmaWNlLnRydXN0ZXEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIB-
CgKCAQEAtwG7AiCEhSD2FoLb0ZDijiKCkJvLFXW7cOMMGfzLHDoux5RpX4
KPVplb4Ngr7zhehs0ccYcoh3g3q4c456HRM67MgdOtI3DPI+sCxL4lJRCPqtC3d
l65VJnkY6BjeCw7KQHoDtuXT9HEM5ir/uuX8+is1ZkRJH1EEBNNHW13Bmk0c
6Wp37sO7TK0qpl5w83kzI4AKfwW5BKrdNvPAzVdjOijDxsUj5u2fH/52QicjYC5+
xe2gb0QanX9fM2UkFehZXnrm+oyavds97rklUkA7Lgj15k0Aqpbvs2C/29pdyKu0
kfTuEywm3+h9SUvAUJyX2OB8KMQQxW+skv0dfcdEQIDAQABo30wezAdBgN
VHQ4EFgQUi7E9wsWY8luRDz6bhP3cX8JBG3QwWgYDVR0RBFMwUYIbbGt
vcHBhcmkub2ZmaWNlLnRydXN0ZXEuY29thjJodHRwczovL2xrb3BwYXJpLm9
mZmljZS50cnVzdGVxLmNvbS9pZHAvc2hpYmJvbGV0aDANBgkqhkiG9w0BA
QUFAAOCAQEAjnn1nbl2cRQBF47BjznFlNONSwthDwz6Mx8NA0LT4knQyDV
Dw3BmOz/yGZaUrShQPVFsetq5xcXXahvEghHtLYYBB9nOARXrbFihlJ6Wi80o
a6AnVD1YCOpeAcMEpCazMP9ebCoT0sYMues9++UkxdGG441LOS0PlE8HjU
XXSbpd/CtD9RQMGJcaNSECQmHep2LX8k53gzLf2unSaOvBVZl22scKW5l1G
CaLvw-
Pave4ZVWYMPRyVuaHP5DFHXo20NcKKOF35X0En7GxrlitzOO+o1wgN31A7
LzUP2sm6OXmDULA3i33cKWSmxnDtXXU03GAACo9MJkpYWrCSjOqdxQ==<
/ds:X509Certificate>
          </ds:X509Data>
        </ds:KeyInfo>
      </ds:Signature>
      <saml2:Subject>
        <saml2:NameID
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
                    NameQualifi-
er="https://lkoppari.office.trusteq.com/idp/shibboleth2"
                    SPNameQualifier="https://sp.testshib.org/shibboleth-sp"
                    >_ff9b01b198ea1ef04100bf6d0ed04d24</saml2:NameID>
        <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
          <saml2:SubjectConfirmationData Address="***.***.***.***"
```

```
                    InResponseTo="_a1a83cbe33543bbecd285286beea3ece"
                                NotOnOrAfter="2015-11-30T08:05:03.077Z"
                                Recipi-
ent="https://sp.testshib.org/Shibboleth.sso/SAML2/POST"/>
         </saml2:SubjectConfirmation>
      </saml2:Subject>
      <saml2:Conditions NotBefore="2015-11-30T08:00:03.077Z"
                  NotOnOrAfter="2015-11-30T08:05:03.077Z"  >
         <saml2:AudienceRestriction>
            <saml2:Audience>https://sp.testshib.org/shibboleth-
sp</saml2:Audience>
         </saml2:AudienceRestriction>
      </saml2:Conditions>
      <saml2:AuthnStatement AuthnInstant="2015-11-30T08:00:03.026Z"
                     SessionIndex="_9cd9d84d4b1956982a1f6bae5c1eef2a">
         <saml2:SubjectLocality Address="***.***.***.***" />
         <saml2:AuthnContext>

<saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Pass
wordProtectedTransport</saml2:AuthnContextClassRef>
         </saml2:AuthnContext>
      </saml2:AuthnStatement>
   </saml2:Assertion>
</saml2p:Response>
```